
Les classes abstraites

L'objectif

Lors de la mise en place de l'héritage, il est possible que certaines fonctionnalités de la classe parent ne soit pas suffisamment complète pour permettre de l'implémenter.

Le concept de méthode abstraite permet d'ajouter la signature de la méthode à la classe parent et de déléguer l'implémentation aux classes enfants.

Attention, une classe ne peut pas être instanciée si elle possède une méthode abstraite.

En python, le module « ABC » permet de fournir l'infrastructure nécessaire pour mettre en place des méthodes abstraites au sein de la classe.

Le module « ABC »

Le module « ABC » (Abstract Base Class) permet de marquer les méthodes ou les propriétés comme abstraites à l'aide du décorateur « `@abc.abstractmethod` »

```
from abc import ABC, abstractmethod

class ClasseAbstraite(ABC):

    @abstractmethod
    def methode_abstraite(self, msg):
        pass

class ClasseEnfant(ClasseAbstraite):

    def methode_abstraite(self, msg):
        print("Afficher : " + msg)
```

Exemple de propriétés abstraites

Propriété à l'aide des méthodes

```
from abc import ABC, abstractmethod

class ClasseAbstraite(ABC):

    @abstractmethod
    def _get_x(self):
        pass

    @abstractmethod
    def _set_x(self, value):
        pass

    x = property(_get_x, _set_x)
```

Propriété à l'aide des décorateurs

```
from abc import ABC, abstractmethod

class ClasseAbstraite(ABC):

    @property
    @abstractmethod
    def msg(self):
        pass

    @msg.setter
    @abstractmethod
    def msg(self, value):
        pass
```